

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Masayuki HIRAYAMA, et al.

GAU:

SERIAL NO: New Application

EXAMINER:

FILED: Herewith

FOR: OBJECT STATE CLASSIFICATION METHOD AND SYSTEM, AND PROGRAM THEREFOR

REQUEST FOR PRIORITY

COMMISSIONER FOR PATENTS
ALEXANDRIA, VIRGINIA 22313

SIR:

☐ Full benefit of the filing date of U.S. Application Serial Number _____, filed _____, is claimed pursuant to the provisions of 35 U.S.C. §120.

☐ Full benefit of the filing date(s) of U.S. Provisional Application(s) is claimed pursuant to the provisions of 35 U.S.C. §119(e):
Application No. _____, Date Filed _____

☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

COUNTRY

Japan

APPLICATION NUMBER

2002-378218

MONTH/DAY/YEAR

December 26, 2002

Certified copies of the corresponding Convention Application(s)

☒ are submitted herewith

☐ will be submitted prior to payment of the Final Fee

☐ were filed in prior application Serial No. _____ filed _____

☐ were submitted to the International Bureau in PCT Application Number _____

Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.

☐ (A) Application Serial No.(s) were filed in prior application Serial No. _____ filed _____; and

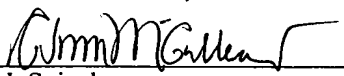
☐ (B) Application Serial No.(s) _____

☐ are submitted herewith

☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.



Marvin J. Spivak

Registration No. 24,913

C. Irvin McClelland
Registration Number 21,124

Customer Number

22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 05/03)

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2002年12月26日

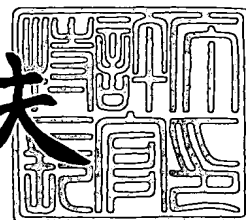
出 願 番 号
Application Number: 特願2002-378218
[ST. 10/C]: [JP2002-378218]

出 願 人
Applicant(s): 株式会社東芝

2003年10月21日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 13B0290291

【提出日】 平成14年12月26日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/00

【発明の名称】 オブジェクト状態分類方法とシステム、およびそのためのプログラム

【請求項の数】 12

【発明者】

 【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
 研究開発センター内

 【氏名】 岡本 渉

【発明者】

 【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
 研究開発センター内

 【氏名】 植木 克彦

【発明者】

 【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
 研究開発センター内

 【氏名】 平山 雅之

【特許出願人】

 【識別番号】 000003078

 【氏名又は名称】 株式会社 東芝

【代理人】

 【識別番号】 100081961

 【弁理士】

 【氏名又は名称】 木内 光春

【手数料の表示】**【予納台帳番号】** 013538**【納付金額】** 21,000円**【提出物件の目録】****【物件名】** 明細書 1**【物件名】** 図面 1**【物件名】** 要約書 1**【プルーフの要否】** 要

【書類名】 明細書

【発明の名称】 オブジェクト状態分類方法とシステム、およびそのためのプログラム

【特許請求の範囲】

【請求項 1】 コンピュータを利用してオブジェクト指向プログラムにおけるオブジェクトの状態を分類する方法において、

判定対象となる判定対象オブジェクトおよびその状態を指定する判定対象オブジェクト指定ステップと、

前記判定対象オブジェクトの前記指定された状態から呼び出されるメソッドを要素とするメソッド集合を作成するメソッド集合作成ステップと、

前記判定対象オブジェクトの前記指定された状態またはその状態からメソッドの呼び出しにより得られた別の状態を判定対象状態とし、その判定対象状態から前記メソッド集合に含まれるメソッドを呼び出すプログラムを実行するプログラム実行ステップと、

前記メソッドを呼び出した場合の実行結果を記録する実行結果記録ステップと、

前記判定対象オブジェクトの前記判定対象状態に対して、前記メソッド集合の要素であるメソッド群と、それらのメソッド群の各メソッドを呼び出した場合の前記実行結果とをマージして擬似状態を作成する擬似状態作成ステップと、を含むことを特徴とするオブジェクト状態分類方法。

【請求項 2】 複数の前記判定対象状態に対して複数の前記擬似状態が作成された場合に、それらの複数の擬似状態を比較して、同一の擬似状態ごとに分類する擬似状態分類ステップを含む、

ことを特徴とする請求項 1 に記載のオブジェクト状態分類方法。

【請求項 3】 前記メソッド集合作成ステップは、

前記メソッド集合として複数のメソッド集合が作成された場合に、使用するメソッド集合を選択するメソッド集合選択ステップを含む、

ことを特徴とする請求項 1 または請求項 2 に記載のオブジェクト状態分類方法。

【請求項 4】 前記メソッド集合作成ステップは、

前記判定対象オブジェクトの前記指定された状態から呼び出されるメソッドを要素とする状態探索用のメソッド集合を作成する第1のメソッド集合作成ステップと、

前記状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた状態から呼び出されるメソッドを要素とする状態判定用のメソッド集合を作成する第2のメソッド集合作成ステップを含み、

前記プログラム実行ステップは、

前記判定対象オブジェクトの前記指定された状態またはその状態からメソッドの呼び出しにより得られた別の状態を実行前状態とし、その実行前状態から前記状態探索用のメソッド集合に含まれるメソッドを呼び出すプログラムを実行する第1の実行ステップと、

前記状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた状態から前記状態判定用のメソッド集合に含まれるメソッドを呼び出すプログラムを実行する第2の実行ステップとを含み、

前記擬似状態作成ステップは、前記状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた状態を前記判定対象状態とし、その判定対象状態に対して、前記状態判定用のメソッド集合の要素であるメソッド群と、それらのメソッド群の各メソッドを呼び出した場合の実行結果とをマージして擬似状態を作成するステップを含む、

ことを特徴とする請求項1乃至請求項3のいずれかに記載のオブジェクト状態分類方法。

【請求項5】 前記擬似状態分類ステップは、

前記判定対象状態に対して得られた前記擬似状態が未登録である場合にその擬似状態を登録するステップを含み、

前記プログラム実行ステップは、

前記擬似状態が新たに登録された場合に、その擬似状態を得た前記判定対象状態を前記実行前状態として、前記第1と第2の実行ステップを繰り返すステップを含む、

ことを特徴とする請求項4に記載のオブジェクト状態分類方法。

【請求項 6】 コンピュータを利用してオブジェクト指向プログラムにおける継承関係にある複数のオブジェクトの状態を分類する方法において、

前記請求項 4 または請求項 5 に記載の方法を用いて、前記複数のオブジェクト中における親クラスと子クラスのオブジェクトの擬似状態を検出する擬似状態検出ステップと、

前記親クラスと子クラスのオブジェクトに対して得られた擬似状態を比較してその差分を検出する擬似状態差分検出ステップとを含み、

前記擬似状態検出ステップは、

前記親クラスのオブジェクトに対し、前記状態探索用と状態判定用のメソッド集合として当該親クラス用の各メソッド集合を使用して、当該親クラスのオブジェクトの擬似状態を検出する親クラス擬似状態検出ステップと、

前記子クラスのオブジェクトに対し、前記状態探索用のメソッド集合として当該親クラス用のメソッド集合とは異なるメソッド集合を使用し、かつ、前記状態判定用のメソッド集合として当該親クラス用のメソッド集合と同じメソッド集合を使用して、当該子クラスのオブジェクトの擬似状態を検出する子クラス擬似状態検出ステップを含む、

ことを特徴とするオブジェクト状態分類方法。

【請求項 7】 コンピュータを利用してオブジェクト指向プログラムにおけるオブジェクトの状態を分類するシステムにおいて、

判定対象となる判定対象オブジェクトおよびその状態を指定する判定対象オブジェクト指定部と、

前記判定対象オブジェクトの前記指定された状態から呼び出されるメソッドを要素とするメソッド集合を作成するメソッド集合作成部と、

前記判定対象オブジェクトの前記指定された状態またはその状態からメソッドの呼び出しにより得られた別の状態を判定対象状態とし、その判定対象状態から前記メソッド集合に含まれるメソッドを呼び出すプログラムを実行するプログラム実行部と、

前記メソッドを呼び出した場合の実行結果を記録する実行結果記録部と、

前記判定対象オブジェクトの前記判定対象状態に対して、前記メソッド集合の

要素であるメソッド群と、それらのメソッド群の各メソッドを呼び出した場合の前記実行結果とをマージして擬似状態を作成する擬似状態作成部と、
を備えたことを特徴とするオブジェクト状態分類システム。

【請求項 8】 前記メソッド集合作成部は、

前記判定対象オブジェクトの前記指定された状態から呼び出されるメソッドを要素とする状態探索用のメソッド集合を作成する第 1 のメソッド集合作成部と、

前記状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた状態から呼び出されるメソッドを要素とする状態判定用のメソッド集合を作成する第 2 のメソッド集合作成部を含み、

前記プログラム実行部は、

前記判定対象オブジェクトの前記指定された状態またはその状態からメソッドの呼び出しにより得られた別の状態を実行前状態とし、その実行前状態から前記状態探索用のメソッド集合に含まれるメソッドを呼び出すプログラムを実行した後に、当該メソッドの呼び出しにより得られた状態から前記状態判定用のメソッド集合に含まれるメソッドを呼び出すプログラムを実行するように構成され、

前記擬似状態作成部は、前記状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた状態を前記判定対象状態とし、その判定対象状態に対して、前記状態判定用のメソッド集合の要素であるメソッド群と、それらのメソッド群の各メソッドを呼び出した場合の実行結果とをマージして擬似状態を作成するように構成された、

ことを特徴とする請求項 7 に記載のオブジェクト状態分類システム。

【請求項 9】 コンピュータを利用してオブジェクト指向プログラムにおける継承関係にある複数のオブジェクトの状態を分類するシステムにおいて、

前記請求項 8 に記載のシステムを用いて、前記複数のオブジェクト中における親クラスと子クラスのオブジェクトの擬似状態を検出する擬似状態検出部と、

前記親クラスと子クラスのオブジェクトに対して得られた擬似状態を比較してその差分を検出する擬似状態差分検出部とを含み、

前記擬似状態検出部は、

前記親クラスのオブジェクトに対し、前記状態探索用と状態判定用のメソッド

集合として当該親クラス用の各メソッド集合を使用して、当該親クラスのオブジェクトの擬似状態を検出すると共に、

前記子クラスのオブジェクトに対し、前記状態探索用のメソッド集合として当該親クラス用のメソッド集合とは異なるメソッド集合を使用し、かつ、前記状態判定用のメソッド集合として当該親クラス用のメソッド集合と同じメソッド集合を使用して、当該子クラスのオブジェクトの擬似状態を検出するように構成された、

ことを特徴とするオブジェクト状態分類システム。

【請求項 10】 コンピュータを利用してオブジェクト指向プログラムにおけるオブジェクトの状態进行分类するためのプログラムにおいて、

判定対象となる判定対象オブジェクトおよびその状態を指定する判定対象オブジェクト指定機能と、

前記判定対象オブジェクトの前記指定された状態から呼び出されるメソッドを要素とするメソッド集合を作成するメソッド集合作成機能と、

前記判定対象オブジェクトの前記指定された状態またはその状態からメソッドの呼び出しにより得られた別の状態を判定対象状態とし、その判定対象状態から前記メソッド集合に含まれるメソッドを呼び出すプログラムを実行するプログラム実行機能と、

前記メソッドを呼び出した場合の実行結果を記録する実行結果記録機能と、

前記判定対象オブジェクトの前記判定対象状態に対して、前記メソッド集合の要素であるメソッド群と、それらのメソッド群の各メソッドを呼び出した場合の前記実行結果とをマージして擬似状態を作成する擬似状態作成機能と、
をコンピュータに実現させることを特徴とするオブジェクト状態分類用のプログラム。

【請求項 11】 前記メソッド集合作成機能は、

前記判定対象オブジェクトの前記指定された状態から呼び出されるメソッドを要素とする状態探索用のメソッド集合を作成する第 1 のメソッド集合作成機能と、

前記状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた

状態から呼び出されるメソッドを要素とする状態判定用のメソッド集合を作成する第2のメソッド集合作成機能を含み、

前記プログラム実行機能は、

前記判定対象オブジェクトの前記指定された状態またはその状態からメソッドの呼び出しにより得られた別の状態を実行前状態とし、その実行前状態から前記状態探索用のメソッド集合に含まれるメソッドを呼び出すプログラムを実行する第1の実行機能と、

前記状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた状態から前記状態判定用のメソッド集合に含まれるメソッドを呼び出すプログラムを実行する第2の実行機能とを含み、

前記擬似状態作成機能は、前記状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた状態を前記判定対象状態とし、その判定対象状態に対して、前記状態判定用のメソッド集合の要素であるメソッド群と、それらのメソッド群の各メソッドを呼び出した場合の実行結果とをマージして擬似状態を作成する機能を含む、

ことを特徴とする請求項10に記載のオブジェクト状態分類用のプログラム。

【請求項12】 コンピュータを利用してオブジェクト指向プログラムにおける継承関係にある複数のオブジェクトの状態を分類するためのプログラムにおいて、

前記請求項11に記載のプログラムを用いて、前記複数のオブジェクト中における親クラスと子クラスのオブジェクトの擬似状態を検出する擬似状態検出機能と、

前記親クラスと子クラスのオブジェクトに対して得られた擬似状態を比較してその差分を検出する擬似状態差分検出機能とをコンピュータに実現させるプログラムであって、

前記擬似状態検出機能は、

前記親クラスのオブジェクトに対し、前記状態探索用と状態判定用のメソッド集合として当該親クラス用の各メソッド集合を使用して、当該親クラスのオブジェクトの擬似状態を検出する親クラス擬似状態検出機能と、

前記子クラスのオブジェクトに対し、前記状態探索用のメソッド集合として当該親クラス用のメソッド集合とは異なるメソッド集合を使用し、かつ、前記状態判定用のメソッド集合として当該親クラス用のメソッド集合と同じメソッド集合を使用して、当該子クラスのオブジェクトの擬似状態を検出する子クラス擬似状態検出機能を含む、
ことを特徴とするオブジェクト状態分類用のプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、オブジェクト指向プログラムにおいて、テストの効率化を目的として擬似的な状態を定義するために使用されるオブジェクト状態分類方法とシステムに関する。

【0002】

【従来の技術】

オブジェクト指向プログラムのクラスに対する単体テストでは、多様な状態のオブジェクトに対して、多様なメソッドを呼び出すという手法が主流である（例えば、特許文献1、2および非特許文献1、2参照。）。この場合、クラスにどのような状態が存在するかはプログラム設計者が決定することであるが、実際に個々のクラスに対してどのような状態が存在するか明確に定義されることは少ない。

【0003】

また、効率的にクラスのテストを行うための具体的な方法としては、例えば、抽象的な状態（メンバ変数の範囲によって状態が定義されている場合など）から具体的な状態を1つだけ作り出してテストに用いる、等の方法も考えられる。しかし、この方法は、状態の定義がなされていて初めて可能となる方法である。逆に言えば、クラスの状態が明確に定義されていない場合には、「クラスの状態」というものに基づいてテストの効率化を図ることは不可能である。

【0004】

【特許文献1】

特開平 9 - 3 2 5 9 5 2 号公報

【特許文献 2】

特開平 7 - 2 6 2 0 4 4 号公報

【非特許文献 1】

ヒョンソクホン (Hyoung Seok Hong) , ヨンレクオン (Young Rae Kwon) , ソントクチャ (Sung Deok Cha) 著, “有限状態機械によるオブジェクト指向テスト (Testing Object-Oriented Based on Finite State Machines) ” , アジア太平洋ソフトウェア工学国際会議 (Asia-Pacific Software Engineering Conference) 発行, 1 9 9 5 年 1 2 月, p. 2 3 4

【非特許文献 2】

ディー・シー・クン (D.C.Kung) , エヌ・スチャック (N.Suchak) , ジェイ・ガオ (J.Gao) , ピー・シーア (P.Hsia) 著, “オブジェクト状態テストについて (On Object State Testing) ” , コンピュータ・ソフトウェア・アプリケーション国際会議 1 9 9 4 (COMPSAC: Computer Software and Applications Conference 1994) , I E E E コンピュータ・ソサイエティ・プレス (IEEE Computer Society Press) 発行, 1 9 9 4 年, p. 2 2 2

【0 0 0 5】

【発明が解決しようとする課題】

前述したように、個々のクラスについて状態の定義が明確になされていない以上、クラスのテストを行う場合のテスターとして、なるべく多数の具体的な状態（各メンバ変数に値を設定したもの。以下、具象状態と呼ぶ）を作り出し、テストを行うことが一般的となる。このようにして作成された多くの具象状態の中には、意味的に同じグループに属すると考えられる複数の具象状態が存在する可能性がある。このことから、設計者が意図する「状態」とは、意味的に等しい具象状態の集合だと考えることもできる。

【0 0 0 6】

しかしながら、意味的に等しい複数の具象状態に対してそれぞれテストを行うことはテスト効率の低下を意味する。この点について、図 1 0 を参照して簡単に説明する。ここで、図 1 0 は、1 つのクラスと、それに対する設計者の意図する

状態の定義の有無とそれに応じたテストケースのリストの一例を示す説明図である。この図10に示すように、設計者が意図する状態が定義されている場合には、「設計者の意図する状態」に該当する「具象状態1」のみからメソッドを呼び出しており、テストケースは2個だけである。これに対し、設計者が意図する状態が定義されていない場合には、想定した全ての具象状態すなわち「具象状態1」、「具象状態2」からメソッドを呼び出しており、テストケースの数は4個と、設計者が意図する状態が定義されている場合に比べて倍増している。

【0007】

このような問題が生じる理由は、クラスの状態がプログラム設計者によって定義されない限り、外部から決定できない点にある。

【0008】

本発明は、上記のような従来技術の課題を解決するために提案されたものであり、その目的は、クラスの状態が明確に定義されていない場合でも、多様な条件におけるプログラムの動作に基づいて擬似的に状態を定義可能とすることにより、クラスの状態の明確な定義の有無に拘わらず、あらゆるクラスに対して、「状態の網羅」という1つの尺度に基づいたテストの効率化が可能なオブジェクト状態分類方法とシステムを提供することである。

【0009】

【課題を解決するための手段】

本発明は、ユーザにより与えられたオブジェクトの具象状態に対して、ユーザにより与えられたメソッド集合中の各メソッドを呼び出すことによって擬似状態を得ることにより、多様な条件におけるプログラムの動作に基づいて擬似的に状態を定義できるようにしたものである。このような本発明によれば、テスト対象となるメソッドを呼び出す前のオブジェクトの具象状態をその擬似状態に応じて限定することにより、テストケースの数を減らしてテストの効率化を実現することができる。

【0010】

請求項1の発明は、コンピュータを利用してオブジェクト指向プログラムにおけるオブジェクトの状態を分類する方法において、判定対象オブジェクト指定ス

テップ、メソッド集合作成ステップ、プログラム実行ステップ、実行結果記録ステップ、擬似状態作成ステップ、を含むことを特徴としている。ここで、判定対象オブジェクト指定ステップは、判定対象となる判定対象オブジェクトおよびその状態を指定するステップであり、メソッド集合作成ステップは、判定対象オブジェクトの指定された状態から呼び出されるメソッドを要素とするメソッド集合を作成するステップである。

【0011】

また、プログラム実行ステップは、判定対象オブジェクトの指定された状態またはその状態からメソッドの呼び出しにより得られた別の状態を判定対象状態とし、その判定対象状態からメソッド集合に含まれるメソッドを呼び出すプログラムを実行するステップであり、実行結果記録ステップは、メソッドを呼び出した場合の実行結果を記録するステップである。さらに、擬似状態作成ステップは、判定対象オブジェクトの判定対象状態に対して、メソッド集合の要素であるメソッド群と、それらのメソッド群の各メソッドを呼び出した場合の実行結果とをマージして擬似状態を作成するステップである。

【0012】

請求項7の発明は、請求項1の発明をシステムの観点から把握したものであり、コンピュータを利用してオブジェクト指向プログラムにおけるオブジェクトの状態を分類するシステムにおいて、請求項1の発明における各ステップに対応する各構成部分として、判定対象オブジェクト指定部、メソッド集合作成部、プログラム実行部、実行結果記録部、擬似状態作成部、を備えたことを特徴としている。

【0013】

請求項10の発明は、請求項1、請求項7の発明をコンピュータプログラムの観点から把握したものであり、コンピュータを利用してオブジェクト指向プログラムにおけるオブジェクトの状態を分類するためのプログラムにおいて、請求項1の発明における各ステップに対応する各機能として、判定対象オブジェクト指定機能、メソッド集合作成機能、プログラム実行機能、実行結果記録機能、擬似状態作成機能、をコンピュータに実現させることを特徴としている。

【0014】

以上のような発明によれば、ユーザにより与えられたオブジェクトの具象状態またはその状態からメソッドの呼び出しにより得られた別の状態を判定対象状態とし、ユーザにより与えられたメソッド集合の各メソッドを判定対象状態から呼び出すことにより、メソッド集合の要素であるメソッドとメソッド呼び出しの実行結果から、判定対象状態の擬似状態を得ることができる。したがって、多様な条件におけるプログラムの動作に基づいて擬似的に状態を定義することができるため、テスト対象となるメソッドを呼び出す前のオブジェクトの具象状態を、得られた擬似状態に応じて限定することにより、テストケースの数を減らしてテストの効率化を実現することができる。

【0015】

請求項2の発明は、請求項1のオブジェクト状態分類方法において、擬似状態分類ステップを含むことを特徴としている。ここで、擬似状態分類ステップは、複数の判定対象状態に対して複数の擬似状態が作成された場合に、それらの複数の擬似状態を比較して、同一の擬似状態ごとに分類するステップである。

【0016】

この発明によれば、同一の擬似状態ごとに分類してグループ化することにより、テスト対象となるメソッドを呼び出す前のオブジェクトの具象状態を、同一の擬似状態となる具象状態ごとにグループ化し、各グループを代表するものだけに絞る等、擬似状態のグループ化を利用して具象状態を適切に限定することができるため、テストケースの数を減らしてテストの効率化を実現することができる。

【0017】

請求項3の発明は、請求項1または請求項2のオブジェクト状態分類方法において、メソッド集合作成ステップが、メソッド集合選択ステップを含むことを特徴としている。ここで、メソッド集合選択ステップは、メソッド集合として複数のメソッド集合が作成された場合に、使用するメソッド集合を選択するステップである。

【0018】

この発明によれば、複数のメソッド集合が作成された場合に、使用するメソッ

ド集合を選択することにより、全てのメソッド集合に含まれる全てのメソッドを呼び出す場合に比べてメソッドの呼び出しを限定することができるため、擬似状態を効率よく作成することができる。

【0019】

請求項4の発明は、請求項1乃至請求項3のいずれかのオブジェクト状態分類方法において、メソッド集合作成ステップが、第1と第2のメソッド集合作成ステップを含み、プログラム実行ステップが、第1と第2の実行ステップを含み、擬似状態作成ステップが、メソッド呼び出しにより得られた判定対象状態の擬似状態作成ステップを含むことを特徴としている。ここで、第1のメソッド集合作成ステップは、判定対象オブジェクトの指定された状態から呼び出されるメソッドを要素とする状態探索用のメソッド集合を作成するステップである。また、第2のメソッド集合ステップは、状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた状態から呼び出されるメソッドを要素とする状態判定用のメソッド集合を作成するステップである。

【0020】

一方、第1の実行ステップは、判定対象オブジェクトの指定された状態またはその状態からメソッドの呼び出しにより得られた別の状態を実行前状態とし、その実行前状態から状態探索用のメソッド集合に含まれるメソッドを呼び出すプログラムを実行するステップである。また、第2の実行ステップは、状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた状態から状態判定用のメソッド集合に含まれるメソッドを呼び出すプログラムを実行するステップである。さらに、メソッド呼び出しにより得られた判定対象状態の擬似状態作成ステップは、状態探索用のメソッド集合に含まれるメソッドの呼び出しにより得られた状態を判定対象状態とし、その判定対象状態に対して、状態判定用のメソッド集合の要素であるメソッド群と、それらのメソッド群の各メソッドを呼び出した場合の実行結果とをマージして擬似状態を作成するステップである。

【0021】

請求項8の発明は、請求項4の発明をシステムの観点から把握したものであり、請求項7のオブジェクト状態分類システムにおいて、メソッド集合作成部が、

請求項 4 の発明における第 1 と第 2 のメソッド集合作成ステップに対応する第 1 と第 2 のメソッド集合作成部を含み、プログラム実行部と擬似状態作成部が、請求項 4 の発明におけるプログラム実行ステップと擬似状態作成ステップに含まれる各ステップに対応する機能を有するように構成されたことを特徴としている。

【0022】

請求項 11 の発明は、請求項 4、請求項 8 の発明をコンピュータプログラムの観点から把握したものであり、請求項 10 のオブジェクト状態分類用のプログラムにおいて、メソッド集合作成機能、プログラム実行機能、および擬似状態作成機能が、請求項 4 の発明におけるメソッド集合作成ステップ、プログラム実行ステップ、および擬似状態作成ステップに含まれる各ステップに対応する各機能を含むことを特徴としている。

【0023】

以上のような発明によれば、ユーザにより与えられた具象状態から、状態探索用のメソッド集合に含まれるメソッドを呼び出して得られた状態を判定対象状態とし、この判定対象状態から、状態判定用のメソッド集合に含まれるメソッドを呼び出すことにより、ユーザにより与えられた具象状態からメソッドを呼び出して得られた状態に対する擬似状態を得ることができる。

【0024】

請求項 5 の発明は、請求項 4 のオブジェクト状態分類方法において、擬似状態分類ステップが、擬似状態を登録するステップを含み、プログラム実行ステップが、第 1 と第 2 の実行ステップを繰り返すステップを含むことを特徴としている。ここで、擬似状態を登録するステップは、判定対象状態に対して得られた擬似状態が未登録である場合にその擬似状態を登録するステップである。また、第 1 と第 2 の実行ステップを繰り返すステップは、擬似状態が新たに登録された場合に、その擬似状態を得た判定対象状態を実行前状態として、第 1 と第 2 の実行ステップを繰り返すステップである。

【0025】

この発明によれば、判定対象状態に対して得られた擬似状態が未登録である場合に擬似状態を登録すると共に、新たな擬似状態が登録された場合にメソッドの

呼び出しを繰り返すことにより、ユーザにより与えられた初期状態に基づいて、オブジェクトが取り得る擬似状態を網羅的に検出することができる。したがって、網羅された擬似状態を得たそれぞれの判定対象状態である各具象状態からテスト対象となるメソッドをそれぞれ呼び出すことにより、網羅的なテストを効率よく行うことができる。

【0026】

請求項6の発明は、コンピュータを利用してオブジェクト指向プログラムにおける継承関係にある複数のオブジェクトの状態を分類する方法において、擬似状態検出ステップと擬似状態差分検出ステップを含むことを特徴としている。ここで、擬似状態検出ステップは、請求項4または請求項5に記載の方法を用いて、複数のオブジェクト中における親クラスと子クラスのオブジェクトの擬似状態を検出するステップである。また、擬似状態差分検出ステップは、親クラスと子クラスのオブジェクトに対して得られた擬似状態を比較してその差分を検出するステップである。

【0027】

さらに、擬似状態検出ステップは、親クラス擬似状態検出ステップと子クラス擬似状態検出ステップを含む。ここで、親クラス擬似状態検出ステップは、親クラスのオブジェクトに対し、状態探索用と状態判定用のメソッド集合として当該親クラス用の各メソッド集合を使用して、当該親クラスのオブジェクトの擬似状態を検出するステップである。また、子クラス擬似状態検出ステップは、子クラスのオブジェクトに対し、状態探索用のメソッド集合として当該親クラス用のメソッド集合とは異なるメソッド集合を使用し、かつ、状態判定用のメソッド集合として当該親クラス用のメソッド集合と同じメソッド集合を使用して、当該子クラスのオブジェクトの擬似状態を検出するステップである。

【0028】

請求項9の発明は、請求項6の発明をシステムの観点から把握したものであり、コンピュータを利用してオブジェクト指向プログラムにおける継承関係にある複数のオブジェクトの状態を分類するシステムにおいて、請求項6の発明における擬似状態検出ステップと擬似状態差分検出ステップに対応する擬似状態検出部

と擬似状態差分検出部を含むことを特徴としている。

【0029】

請求項12の発明は、請求項6、請求項9の発明をコンピュータプログラムの観点から把握したものであり、請求項6の発明における擬似状態検出ステップと擬似状態差分検出ステップに対応する擬似状態検出機能と擬似状態差分検出機能をコンピュータに実現させることを特徴としている。

【0030】

以上のような発明によれば、子クラスのオブジェクトに対して、子クラス独自の状態探索用のメソッド集合を使用し、かつ、親クラスと共通の状態判定用のメソッド集合を使用して、擬似状態を検出し、親クラスのオブジェクトに対して得られた擬似状態と比較することにより、子クラスが親クラスに対して不正な状態を作り出しているか否かを容易にチェックすることができる。

【0031】

【発明の実施の形態】

以下には、本発明の実施形態を図面に沿って具体的に説明する。ただし、ここで記載する実施形態は、本発明を何ら限定するものではなく、本発明の一態様を例示するものにすぎない。

【0032】

本発明は、典型的には、コンピュータをソフトウェアで制御することにより実現される。この場合のソフトウェアは、コンピュータのハードウェアを物理的に活用することで本発明の作用効果を実現するものであり、また、従来技術を適用可能な部分には好適な従来技術が適用される。さらに、本発明を実現するハードウェアやソフトウェアの具体的な種類や構成、ソフトウェアで処理する範囲などは自由に変更可能であり、例えば、本発明を実現するプログラムは本発明の一態様である。

【0033】

[1. 用語の説明]

まず、以下の実施形態において使用する用語のうち、「具象状態」、「状態判定メソッド集合」、「擬似状態」、「状態探索メソッド集合」、について次のよ

うに定義する。

【0034】

[1-1. 具象状態]

「具象状態」は、「クラスの各メンバ変数を取る値を具体的に指定した集合」を意味している。以下の表1は、具象状態の一例を示している。

【表1】

```
具象状態の例 {  
    angle = 30,  
    flgLock = true  
}
```

【0035】

[1-2. 状態判定メソッド集合]

「状態判定メソッド集合」は、「メソッドと引数の値の組を要素とする集合」を意味している。なお、メソッド内でグローバル変数が参照されている場合には、その値も含む。また、1つの要素は、メソッド1つだけである必要はなく、複数のメソッドからなるメソッド系列でもよい。下の表2に示す例では、状態判定メソッド集合の3番目の要素のみがメソッド系列となっている。このメソッド系列は、与えられた状態から、「lock()メソッドを呼び出し、引き続いてisOpened()メソッドを呼び出す」ことを意味している。

【表2】

```
状態判定メソッド集合の例 {  
    {メソッド:setAngle, 引数:a=10},  
    {メソッド:setAngle, 引数:a=45},  
    {メソッド:lock, 引数(なし), メソッド:isOpened, 引数(なし)},  
    {メソッド:isOpened, 引数(なし)}  
}
```

【0036】

[1-3. 擬似状態]

「擬似状態」は、「状態判定メソッド集合の各要素と、それに対する実行結果の組を要素とする集合」を意味している。下の表3に示す例では、実行結果としてパスを用いている。

【表 3】

擬似状態の例 {

```
{ {メソッド:setAngle, 引数:a=10}, 実行結果:Path01},  
{ {メソッド:setAngle, 引数:a=45}, 実行結果:Path02},  
{ {メソッド:lock, 引数(なし), メソッド:isOpen, 引数(なし)}, 実行結果:Path03},  
{ {メソッド:isOpen, 引数(なし)}, 実行結果:Path04}
```

}

【0037】

[1-4. 状態探索メソッド集合]

「状態探索メソッド集合」は、基本的に、「状態判定メソッド集合」と同様の集合を意味しているが、メソッド系列を1つの要素として指定しない点で異なる。

【0038】

[2. システム構成]

図1は、本発明を適用した一つの実施形態に係るオブジェクト状態分類システムの構成を示すブロック図である。この図1に示すように、本実施形態のオブジェクト状態分類システムは、判定対象オブジェクト指定部101、状態探索メソッド集合作成部102、状態判定メソッド集合作成部103、状態判定メソッド集合選択部104、プログラム実行部105、実行結果記録部106、擬似状態作成部107、擬似状態分類部108、擬似状態差分検出部109、ソースコード解析部110、等を備えている。各部の詳細は次の通りである。

【0039】

判定対象オブジェクト指定部101は、ユーザの指定に応じて、判定対象となる判定対象オブジェクトを指定すると共に、その判定対象オブジェクトの判定対象となる判定対象状態あるいは初期状態を指定する部分である。

【0040】

状態探索メソッド集合作成部102は、ユーザの指定に応じて、判定対象オブジェクトの指定された初期状態から呼び出されるメソッドを要素とする状態探索メソッド集合を作成して状態探索メソッド集合データベース111に登録する部分であり、本発明における第1のメソッド集合作成部に相当する。

【0041】

状態判定メソッド集合作成部 103 は、ユーザの指定に応じて、状態探索メソッド集合データベース 111 に登録された状態探索メソッド集合に含まれるメソッドの呼び出しにより得られた状態から呼び出されるメソッドを要素とする状態判定メソッド集合を作成して状態判定メソッド集合データベース 112 に登録する部分であり、本発明における第 2 のメソッド集合作成部に相当する。

【0042】

状態判定メソッド集合選択部 104 は、状態判定メソッド集合作成部 103 により複数の状態判定メソッド集合が作成されて状態判定メソッド集合データベース 112 に登録された場合に、使用する状態判定メソッド集合を選択する部分である。

【0043】

プログラム実行部 105 は、オブジェクトのある状態からメソッドを呼び出す実行プログラム 113 を実行する部分であり、特に、判定対象オブジェクト指定部 101 により指定された状態またはその状態からメソッドの呼び出しにより得られた別の状態を判定対象状態とし、その判定対象状態からメソッド集合に含まれるメソッドを呼び出す部分である。すなわち、プログラム実行部 105 は、実行プログラム 113 の実行により、後述する「状態の分類」動作時には、判定対象オブジェクト指定部 101 により指定された判定対象状態から判定対象メソッド集合に含まれるメソッドを呼び出すようになっている。

【0044】

プログラム実行部 105 はまた、後述する「擬似状態の検出」動作時には、実行プログラム 113 の実行により、指定された初期状態から状態探索メソッド集合に含まれるメソッドを呼び出し、続いて状態判定メソッド集合に含まれるメソッドを呼び出すようになっている。すなわち、プログラム実行部 105 は、「擬似状態の検出」動作時には、指定された初期状態またはその初期状態からメソッドの呼び出しにより得られた別の状態を実行前状態とし、その実行前状態から状態探索メソッド集合に含まれるメソッドを呼び出し、続いて、そのメソッド呼び出しにより得られた状態から状態判定メソッド集合に含まれるメソッドを呼び出すようになっている。

【0045】

実行結果記録部106は、状態探索メソッド集合または状態判定メソッド集合に含まれるメソッドを呼び出した場合の実行結果を記録する部分である。

【0046】

擬似状態作成部107は、判定対象オブジェクトの判定対象状態に対して、判定対象メソッド集合の要素であるメソッド群と、それらのメソッド群の各メソッドを呼び出した場合の実行結果とをマージして擬似状態を作成する部分である。

【0047】

擬似状態分類部108は、判定対象オブジェクトの各判定対象状態に対して得られた擬似状態を比較して、同一の擬似状態ごとに分類する部分である。そして、後述する「状態の分類」動作時においては、複数の判定対象状態を、同一の擬似状態となる判定対象状態ごとにグループ化するようにになっている。擬似状態分類部108はまた、後述する「擬似状態の検出」動作時においては、得られた擬似状態を登録済みの擬似状態と比較して、得られた擬似状態が未登録である場合にその擬似状態を新規に検出された擬似状態として擬似状態データベース114に登録するようになっている。

【0048】

擬似状態差分検出部109は、継承関係にある複数のオブジェクトの状態を分類する場合に、判定対象オブジェクト指定部101～擬似状態分類部108までの各部の動作により親クラスと子クラスのオブジェクトに対してそれぞれ得られた擬似状態を比較してその差分を検出する部分である。

【0049】

ソースコード解析部110は、ソースコードデータベース115に格納された分類対象となるオブジェクト指向プログラムのソースコードを解析して、判定対象オブジェクト指定部101、状態探索メソッド集合作成部102、状態判定メソッド集合作成部103、等の部分に必要なデータを提供する部分である。

【0050】**[3. 動作]**

以下には、以上のような構成を有する本実施形態のオブジェクト状態分類シス

テムによる具体的な動作について説明する。

【0051】

[3-1. 状態の分類]

図2は、本実施形態のオブジェクト状態分類システムの基本的な動作として、ユーザにより指定されたオブジェクトの具象状態に対する擬似状態を取得し、得られた擬似状態に応じて具象状態を分類する動作の概略を示すフローチャートである。また、図3は、図2に示す「状態の分類」動作の対象となる具体的なオブジェクト指向プログラムのプログラムリストの一例を示す説明図である。

【0052】

以下には、これらの図2、図3を参照しながら、本実施形態のオブジェクト状態分類システムによる状態の分類について具体的に説明する。なお、本明細書中において、ユーザによる「指定」は、具体的には、「コンピュータに一般的に付随または接続された入出力装置によるユーザの指定操作」を意味しており、判定対象オブジェクト指定部101等による「指定」は、「コンピュータによるデータの指定」を意味している。

【0053】

図2に示すように、状態の分類においてはまず、ユーザによる指定に応じて、判定対象オブジェクト指定部101により、判定対象となる判定対象オブジェクトを指定すると共に、その判定対象オブジェクトの判定対象となる判定対象状態を指定する(S201)。すなわち、どのクラスのインスタンスを判定対象オブジェクトとし、また、そのインスタンスのどの具象状態を判定対象状態とするかを指定する。ここでは例えば、図3に示すクラス「Door」のインスタンスの具象状態として、以下の表4に示す8個の具象状態「状態01」～「状態08」を指定したものとする。また、子クラスのインスタンスに対して、親クラスとしての具象状態を指定することも考えられる。

【0054】

【表 4】

状態 01	angle = 0	flgLock = false
状態 02	angle = 0	flgLock = true
状態 03	angle = 45	flgLock = false
状態 04	angle = 45	flgLock = true
状態 05	angle = 60	flgLock = false
状態 06	angle = 60	flgLock = true
状態 07	angle = 90	flgLock = false
状態 08	angle = 90	flgLock = true

【0 0 5 5】

次に、ユーザによる指定に応じて、状態判定メソッド集合作成部 1 0 3 により、擬似状態の判定、分類を行うために必要となるメソッドを要素とする状態判定メソッド集合を作成して状態判定メソッド集合データベース 1 1 2 に登録する（S 2 0 2）。この状態判定メソッド集合の作成は、メソッド集合の設定、参照する引数の設定、およびグローバル変数の設定を含む。また、状態判定メソッド集合を複数準備しておき、プログラムを実行する際に、その複数の状態判定メソッド集合の中から使用する状態判定メソッド集合を選択することも可能である。

【0 0 5 6】

ここでは例えば、下の表 5 に示すように、図 3 に示すクラス「Door」に定義されている全てのメソッドを状態判定メソッドとして使用し、さらに、「setAngle」メソッドについては、引数の値として、「-1 0」、「4 5」、「1 0 0」という 3 つの値を与えたものとする。

【0 0 5 7】

【表 5】

```

状態判定メソッド集合 JudgeSet1 {
    {メソッド: Door::isOpen, 引数:(なし)},
    {メソッド: Door::unlock, 引数:(なし)},
    {メソッド: Door::lock, 引数:(なし)},
    {メソッド: Door::setAngle, 引数:a=-10 },
    {メソッド: Door::setAngle, 引数:a=45 },
    {メソッド: Door::setAngle, 引数:a=100 }
}

```

【0 0 5 8】

なお、状態判定メソッド集合データベース 112 に複数の状態判定メソッド集合が登録されている場合には、ユーザによる指定に応じて、状態判定メソッド集合選択部 104 により、どの状態判定メソッド集合を用いて擬似状態の判定を行うか選択する。また、状態判定メソッド集合が 1 つしか存在しない場合には、その状態判定メソッド集合が自動的に選択される。

【0059】

続いて、プログラム実行部 105 により、判定対象オブジェクトの指定された各具象状態に対して、選択された状態判定メソッド集合に含まれる全てのメソッドをそれぞれ呼び出す (S203)。ここで注意すべきは、状態判定メソッドを呼び出す前のオブジェクトは、必ず指定された具象状態でなければならない点である。なぜなら、状態判定メソッド集合に含まれるメソッドを連続的に呼び出した場合には、あるメソッドがオブジェクトの状態を変えてしまう可能性があるからである。したがって、状態判定メソッドがオブジェクトの状態を変えないことが保証されていない限り、状態判定メソッドを呼び出す前には、オブジェクトを指定の具象状態になるようにセットしておく必要がある。

【0060】

次に、実行結果記録部 106 により、プログラム実行部 105 によるプログラムの実行結果を記録する (S204)。この場合、「実行結果」としては、具体的には、「メソッドの戻り値」、「実行パス」等が考えられる。この例では、実行結果として、呼び出したメソッド内部での実行パスを用いるものとする。なお、この場合の「実行パス」とは、分岐文において「true」であったか、「false」であったか等、分岐においてどれを実行したかについての記録を意味する。また、図 3 に示すプログラム内においては、コメントで「ブロック 11」等と記入しているが、このブロック番号の列が実行パスに相当する。

【0061】

続いて、擬似状態作成部 107 により、判定対象状態である各具象状態に対して、プログラムの実行に使用した状態判定メソッド集合と実行結果とを足して擬似状態を作成し、擬似状態データベース 114 に登録する (S205)。下の表 6 は、表 4 に示した具象状態「状態 01」に対して表 5 に示した状態判定メソッ

ド集合を使用してプログラムを実行した場合に作成された擬似状態を示している。この例では、実行パスとして、図3のプログラム中に記述されているブロック番号を用いている。なお、この例では、実行パスが単一のブロック番号となっているが、一般的には、実行パスはブロック番号の列となる。

【0062】

【表6】

状態 01 に対する擬似状態 {

```
{ {メソッド: Door::isOpened, 引数:(なし)}, 実行パス:ブロック 41},
{ {メソッド: Door::unLock, 引数:(なし)}, 実行パス:ブロック 31},
{ {メソッド: Door::lock, 引数:(なし)}, 実行パス:ブロック 22},
{ {メソッド: Door::setAngle, 引数:a=-10 }, 実行パス:ブロック 12},
{ {メソッド: Door::setAngle, 引数:a=45 }, 実行パス:ブロック 13},
{ {メソッド: Door::setAngle, 引数:a=100 }, 実行パス:ブロック 14}
}
```

【0063】

また、表4に示した残り7個の具象状態「状態02」～「状態08」に対しても、同様に擬似状態が作成される。下の表7は、8個の具象状態「状態01」～「状態08」に対してそれぞれ作成された擬似状態を示している。

【0064】

【表7】

状態名	isOpened 引数:なし	unLock 引数:なし	lock 引数:なし	setAngle 引数:-10	setAngle 引数:45	setAngle 引数:100	擬似状態名
状態 01	41	31	22	12	13	14	擬似状態 A
状態 02	41	31	22	11	11	11	擬似状態 B
状態 03	42	31	21-22	12	13	14	擬似状態 C
状態 04	42	31	21-22	11	11	11	擬似状態 D
状態 05	42	31	21-22	12	13	14	擬似状態 C
状態 06	42	31	21-22	11	11	11	擬似状態 D
状態 07	42	31	21-22	12	13	14	擬似状態 C
状態 08	42	31	21-22	11	11	11	擬似状態 D

【0065】

以上のようにして判定対象オブジェクトの各具象状態に対して、それぞれ擬似状態が定義されるが、判定対象状態である異なる具象状態に対して、擬似状態が同一となる可能性がある。そこで、本実施形態においては、擬似状態分類部10

8 により、得られた複数の擬似状態を比較して、同一の擬似状態ごとに分類してグループ化し、それらの擬似状態を得た複数の異なる具象状態を、同一の擬似状態となる具象状態ごとにグループ化する（S 2 0 6）。

【0 0 6 6】

上記の表 7 に示す例においては、8 個の具象状態「状態 0 1」～「状態 0 8」に対して 4 つの擬似状態「擬似状態 A」～「擬似状態 D」のいずれかが得られているため、下の表 8 に示すように、8 個の具象状態「状態 0 1」～「状態 0 8」を、得られた擬似状態「擬似状態 A」～「擬似状態 D」ごとに 4 つのグループに分類することができる。

【0 0 6 7】

【表 8】

擬似状態名	状態
擬似状態 A	状態 01
擬似状態 B	状態 02
擬似状態 C	状態 03, 05, 07
擬似状態 D	状態 04, 06, 08

【0 0 6 8】

[3 - 2 . 状態の分類による作用効果]

以上のような本実施形態のオブジェクト状態分類システムによる「状態の分類」によれば、ユーザにより与えられたオブジェクトの具象状態を判定対象状態とし、ユーザにより与えられた状態判定メソッド集合の各メソッドを判定対象状態から呼び出すことにより、状態判定メソッド集合の要素であるメソッド群とメソッド呼び出しの実行結果から、判定対象状態の擬似状態を得ることができる。

【0 0 6 9】

したがって、多様な条件におけるプログラムの動作に基づいて擬似的に状態を定義することができるため、テスト対象となるメソッドを呼び出す前のオブジェクトの具象状態を、得られた擬似状態に応じて限定することにより、テストケースの数を減らしてテストの効率化を実現することができる。

【0 0 7 0】

特に、同一の擬似状態となる具象状態ごとにグループ化することにより、メソ

ッドを呼び出す前のテスト対象となるオブジェクトの具象状態を、各グループを代表するものだけに絞る等、擬似状態によるグループ化を利用して具象状態を適切に限定することができるため、テストケースの数を減らしてテストの効率化を実現することができる。この点について、以下により詳細に説明する。

【0071】

まず、上記のような「状態の分類」を用いた実用上の手法としては、例えば、多様なオブジェクトの具象状態からメソッドを呼び出すテストを行おうとする場合に、予め擬似状態により具象状態群を分類しておくことが考えられる。図3、表4～表8に示した例では、例えば、図4に示すように、4個の擬似状態「擬似状態A」～「擬似状態D」を代表する具象状態を1つずつ選び出し、その状態のみをテストすることが考えられる。この場合には、本来は8個の具象状態「状態01」～「状態08」に対してテストを行う予定であったのに対し、各擬似状態を代表する4個の具象状態「状態01」、「状態02」、「状態05」、「状態06」のみに対してテストを行うことにより、テストケースの数を半分に減らしてテストの効率化を図ることができる。

【0072】

また、上記のような状態の分類によれば、複数の状態判定メソッド集合が作成された場合に、状態判定メソッド集合選択部104により、使用する状態判定メソッド集合を選択することにより、全ての状態判定メソッド集合に含まれる全てのメソッドを呼び出す場合に比べてメソッドの呼び出しを限定することができるため、擬似状態を効率よく作成することができる。

【0073】

[3-3. 擬似状態の検出]

図5は、本実施形態のオブジェクト状態分類システムによる別の動作として、ユーザにより指定されたオブジェクトの初期状態からメソッドを呼び出して得られる具象状態に対する擬似状態を検出する動作の概略を示すフローチャートである。この擬似状態の検出においては、初期状態から状態探索メソッドを呼び出し、それにより得られる具象状態に対して状態判定メソッドの呼び出しにより擬似状態を検出し、検出した擬似状態が未登録の擬似状態である場合には、さらに状

状態探索メソッドを呼び出し、それにより得られる具象状態に対して状態判定メソッドの呼び出しにより擬似状態を検出する、という手順を繰り返すものである。以下には、この図5を参照しながら、本実施形態のオブジェクト状態分類システムによる擬似状態の検出について具体的に説明する。

【0074】

図5に示すように、擬似状態の検出においてはまず、ユーザによる指定に応じて、判定対象オブジェクト指定部101により、判定対象となる判定対象オブジェクトとその初期状態を指定する（S501）。具体的には、特定のコンストラクタを指定することで代替できる。

【0075】

次に、ユーザによる指定に応じて、状態探索メソッド集合作成部102により、初期状態から呼び出されるメソッドを要素とする状態探索メソッド集合を作成して状態探索メソッド集合データベース111に登録する（S502）。この状態探索メソッド集合の作成は、メソッド集合の設定、参照する引数の設定、およびグローバル変数の設定を含む。

【0076】

なお、あるメソッドが状態探索メソッド集合に含まれることの意味は、「そのメソッドを呼び出した後の状態を擬似状態として判定する」ことにある。したがって、状態探索メソッド集合に含まれるメソッドは、本来、オブジェクトの状態を変更する可能性があるものだけにすることが望ましい。

【0077】

続いて、ユーザによる指定に応じて、状態判定メソッド集合作成部103により、擬似状態の判定、分類を行うために必要となるメソッドを要素とする状態判定メソッド集合を作成して状態判定メソッド集合データベース112に登録する（S503）。この状態判定メソッド集合の作成・登録については、「状態の分類」動作における状態判定メソッド集合の作成・登録について説明した通りである。

【0078】

次に、指定された初期状態またはメソッド呼び出し後の状態を実行前状態とし

て、プログラム実行部 105 により、その実行前状態に対して、状態探索メソッド集合に含まれる全てのメソッドと状態判定メソッド集合に含まれる全てのメソッドとの全ての組み合わせをそれぞれ呼び出す。すなわち、実行前状態から状態探索メソッド集合に含まれる 1 個のメソッドを呼び出し、続いて、そのメソッド呼び出しにより得られた状態を判定対象状態として、その判定対象状態から状態判定メソッド集合に含まれる 1 個のメソッドを呼び出す (S504)。

【0079】

図 6 は、このようなプログラム実行手順の一例を示す説明図である。この図 6 の例においては、初期状態「初期状態 α 」が指定されると共に、3 個のメソッド「method01()」、「method02()」、「method03()」を要素とする状態探索メソッド集合が作成され、3 個のメソッド「judge01()」、「judge02()」、「judge03()」を要素とする状態判定メソッド集合が作成されている場合を示している。

【0080】

この図 6 の例では、「初期状態 α 」から状態探索メソッド集合に含まれるメソッド「method01()」を呼び出した後の状態を「状態 α' 」としている。そして、この状態「状態 α' 」から状態判定メソッド集合に含まれる 3 個のメソッド「judge01()」、「judge02()」、「judge03()」をそれぞれ呼び出している。同様に、「初期状態 α 」から「method02()」を呼び出した後の状態を「状態 α'' 」、「初期状態 α 」から「method03()」を呼び出した後の状態を「状態 α''' 」として、それぞれの状態から 3 個のメソッド「judge01()」、「judge02()」、「judge03()」をそれぞれ呼び出す。この例では、下の表 9 に示すように、合計 9 個の実行パターンが存在する。なお、表中の矢印「→」は、連続してメソッドを呼び出すことを意味している。

【0081】

【表 9】

1. 初期状態 $\alpha \rightarrow \text{method01}() \rightarrow \text{judge01}()$
2. 初期状態 $\alpha \rightarrow \text{method01}() \rightarrow \text{judge02}()$
3. 初期状態 $\alpha \rightarrow \text{method01}() \rightarrow \text{judge03}()$
4. 初期状態 $\alpha \rightarrow \text{method02}() \rightarrow \text{judge01}()$
5. 初期状態 $\alpha \rightarrow \text{method02}() \rightarrow \text{judge02}()$
6. 初期状態 $\alpha \rightarrow \text{method02}() \rightarrow \text{judge03}()$
7. 初期状態 $\alpha \rightarrow \text{method03}() \rightarrow \text{judge01}()$
8. 初期状態 $\alpha \rightarrow \text{method03}() \rightarrow \text{judge02}()$
9. 初期状態 $\alpha \rightarrow \text{method03}() \rightarrow \text{judge03}()$

【0 0 8 2】

次に、実行結果記録部 1 0 6 により、プログラム実行部 1 0 5 によるプログラムの実行結果を記録する（S 5 0 5）。このような実行結果の記録については、「状態の分類」動作における実行結果の記録について説明した通りである。

【0 0 8 3】

続いて、擬似状態作成部 1 0 7 により、判定対象状態である各具象状態に対して、プログラムの実行に使用した状態判定メソッド集合と実行結果とを足して擬似状態を作成する（S 5 0 6）。このような擬似状態の作成・登録については、「状態の分類」動作における擬似状態の作成について説明した通りである。

【0 0 8 4】

続いて、擬似状態分類部 1 0 8 により、得られた複数の擬似状態を比較して、同一の擬似状態ごとに分類してグループ化し、各グループの擬似状態を登録済みの擬似状態と比較して同じものが存在しない場合、すなわち、得られた擬似状態が未登録である場合には、その擬似状態を新規に検出された擬似状態として擬似状態データベース 1 1 4 に登録する（S 5 0 7）。

【0 0 8 5】

そして、新たに登録された擬似状態がある場合（S 5 0 8 の YES）には、その登録された擬似状態を得た具象状態を実行前状態として、プログラム実行部 1 0 5 によるメソッド呼び出し、実行結果記録部 1 0 6 によるその実行結果の記録、擬似状態作成部 1 0 7 による擬似状態の作成、擬似状態分類部 1 0 8 による擬似状態の分類・登録、という一連の手順（S 5 0 4 ～ S 5 0 7）を繰り返す。

【0 0 8 6】

図7は、このような一連の手順（S504～S507）を行った場合の処理内容の一例を示す説明図である。この図7に示す例においては、初期状態として、コンストラクタ「constructor」が指定され、このコンストラクタ「constructor」から得られる擬似状態が「擬似状態A」として登録されている場合に、一連の手順（S504～S507）を5回繰り返した場合を示している。

【0087】

すなわち、この例では、第1回目の処理として、初期状態から「method01()」、「method02()」、「method03()」をそれぞれ呼び出して得られた各状態を判定対象状態とし、それらの判定対象状態に対して、登録済みの「擬似状態A」以外に、未登録の擬似状態「擬似状態B」が得られているため、その「擬似状態B」を登録し、「擬似状態B」を得た具象状態から第2回目の処理を行っている。

【0088】

この第2回目の処理においては、未登録の「擬似状態C」、「擬似状態D」が得られているため、「擬似状態C」を登録し、「擬似状態C」を得た具象状態からの第3回目の処理を行うと共に、「擬似状態D」を登録し、「擬似状態D」を得た具象状態から第4回目の処理を行っている。この場合、第3回目の処理においては、未登録の擬似状態が得られないが、第4回目の処理においては、未登録の「擬似状態E」が得られているため、「擬似状態E」を登録し、「擬似状態E」を得た具象状態から第5回目の処理を行っている。この第5回目の処理においては、未登録の擬似状態が得られないため、繰り返し処理を終了している。

【0089】

このように、新たな擬似状態が登録される限り（S508のNO）、一連の手順（S504～S507）を何回でも繰り返して、擬似状態の検出を続ける。そして、最終的に、新たな擬似状態が登録されなくなった時点（S508のNO）で、「擬似状態の検出」を終了する。

【0090】

〔3-4. 擬似状態の検出による作用効果〕

以上のような本実施形態のオブジェクト状態分類システムによる「擬似状態の検出」によれば、ユーザにより与えられた具象状態から、状態探索メソッド集合

に含まれるメソッドを呼び出して得られた状態を判定対象状態とし、この判定対象状態から、状態判定メソッド集合に含まれるメソッドを呼び出すことにより、ユーザにより与えられた具象状態からメソッドを呼び出して得られた状態に対する擬似状態を得ることができる。

【0091】

特に、判定対象状態に対して得られた擬似状態が未登録である場合に擬似状態を登録すると共に、新たな擬似状態が登録された場合にメソッドの呼び出しを繰り返すことにより、ユーザにより与えられた初期状態に基づいて、オブジェクトが取り得る擬似状態を網羅的に検出することができる。したがって、網羅された擬似状態を得たそれぞれの判定対象状態である各具象状態からテスト対象となるメソッドをそれぞれ呼び出すことにより、網羅的なテストを効率よく行うことができる。

【0092】

[3-5. クラス階層内での擬似状態の差分検出]

図8は、本実施形態のオブジェクト状態分類システムによる別の動作として、前述した「擬似状態の検出」を用いて、クラス階層内での擬似状態の差分を検出する動作の概略を示すフローチャートである。また、図9は、図8に示す「クラス階層内での擬似状態の差分検出」の対象となる継承関係にある親クラスと子クラス、およびそれらのクラスにおける擬似状態の検出結果の一例を示す説明図である。以下には、これらの図8、図9を参照しながら、本実施形態のオブジェクト状態分類システムによる「クラス階層内での擬似状態の差分検出」について具体的に説明する。

【0093】

まず、親クラスの判定対象オブジェクトに対し、親クラス用の状態探索メソッド集合と状態判定メソッド集合を用いて、図5に示すような「擬似状態の検出」を行い、親クラスのオブジェクトが取る擬似状態を検出する(S801)。図9の例では、親クラスの判定対象オブジェクトとしてインスタンス「Parent」を指定し、その初期状態として「p1=0, p2=0」を指定している。そして、この初期状態から、親クラスのメソッド群「PMethod01()」、「PMethod02()」、「PMethod0

3()」を、親クラス用の状態探索メソッド集合および状態判定メソッド集合として用いて、親クラスのオブジェクトが取る擬似状態を検出している。この図9の例では、親クラスのオブジェクトが取る擬似状態として、「擬似状態A」、「擬似状態B」という2個の擬似状態が検出されている。

【0094】

次に、子クラスの判定対象オブジェクトに対し、子クラス用の状態探索メソッド集合と、親クラス用の状態判定メソッド集合を用いて、図5に示すような「擬似状態の検出」を行い、子クラスのオブジェクトが取る、「親クラスとしての」擬似状態を検出する(S802)。図9の例では、子クラスの判定対象オブジェクトとしてインスタンス「Child」を指定し、その初期状態として「p1=0, p2=0」を指定している。そして、この初期状態から、子クラスのメソッド群「CMethod01()」、「CMethod02()」を、子クラス用の状態探索メソッド集合として用いると共に、親クラスと同じ状態判定メソッド集合を用いて、子クラスのオブジェクトが取る擬似状態を検出している。この図9の例では、子クラスのオブジェクトが取る擬似状態として、「擬似状態A」、「擬似状態B」、「擬似状態C」という3個の擬似状態が検出されている。

【0095】

続いて、親クラスと子クラスのオブジェクトに対して得られた擬似状態を比較してその差分を検出する(S803)。すなわち、子クラスのオブジェクトに対して検出された擬似状態の中に、親クラスのオブジェクトに対して検出されていない擬似状態があるか否かを検出する。図9の例においては、子クラスのオブジェクトに対して検出された擬似状態の中に、親クラスのオブジェクトに対して検出されていない「擬似状態C」が存在するため、この「擬似状態C」が差分として検出される。

【0096】

[3-6. クラス階層内での擬似状態の差分検出による作用効果]

以上のような本実施形態のオブジェクト状態分類システムによる「クラス階層内での擬似状態の差分検出」によれば、子クラスのオブジェクトに対して、子クラス独自の状態探索用のメソッド集合を使用し、かつ、親クラスと共通の状態判

定用のメソッド集合を使用して、擬似状態を検出し、親クラスのオブジェクトに対して得られた擬似状態と比較することにより、子クラスが親クラスに対して未発見の状態を作り出しているか否かを容易にチェックすることができる。

【0097】

したがって、子クラスのオブジェクトに対して検出された擬似状態の中に、親クラスのオブジェクトに対して検出されていない擬似状態が存在する場合には、子クラスが、親クラスとして不正な、あるいは、チェックしていない状態を作り出した可能性があるものとして、ユーザに何らかの警告を発する等により、ユーザの注意を喚起することができる。

【0098】

[4. 他の実施形態]

なお、本発明は、前述した実施形態に限定されるものではなく、本発明の範囲内で他にも多種多様な形態が実施可能である。

【0099】

例えば、図1に示したシステム構成や、図2、図5、図8等にした処理手順は、一例にすぎず、具体的なシステム構成や処理手順は自由に変更可能である。また、図3に示したプログラムリストは一例にすぎず、本発明は、各種のオブジェクト指向プログラムに同様に適用可能であり、同様に優れた効果が得られるものである。

【0100】

【発明の効果】

以上説明したように、本発明によれば、ユーザにより与えられたオブジェクトの具象状態に対して、ユーザにより与えられたメソッド集合中の各メソッドを各状態から呼び出すことによって擬似状態を得ることにより、多様な条件におけるプログラムの動作に基づいて擬似的に状態を定義することができる。したがって、クラスの状態の明確な定義の有無に拘わらず、あらゆるクラスに対して、「状態の網羅」という1つの尺度に基づいたテストの効率化が可能なオブジェクト状態分類方法とシステムを提供することができる。

【図面の簡単な説明】

【図 1】

本発明を適用した一つの実施形態に係るオブジェクト状態分類システムの構成を示すブロック図。

【図 2】

図 1 に示すオブジェクト状態分類システムの基本的な動作として、「状態の分類」動作の概略を示すフローチャート。

【図 3】

図 2 に示す「状態の分類」動作の対象となる具体的なオブジェクト指向プログラムのプログラムリストの一例を示す説明図。

【図 4】

図 2 に示す「状態の分類」を用いた実用上の手法の一例を示す説明図。

【図 5】

図 1 に示すオブジェクト状態分類システムによる「擬似状態の検出」動作の概略を示すフローチャート。

【図 6】

図 5 に示す「擬似状態の検出」動作におけるプログラム実行手順の一例を示す説明図。

【図 7】

図 5 に示す「擬似状態の検出」動作において、実行前状態からの一連の手順を行った場合の処理内容の一例を示す説明図。

【図 8】

図 1 に示すオブジェクト状態分類システムによる「クラス階層内での擬似状態の差分検出」動作の概略を示すフローチャート。

【図 9】

図 8 に示す「クラス階層内での擬似状態の差分検出」の対象となる継承関係にある親クラスと子クラス、およびそれらのクラスにおける擬似状態の検出結果の一例を示す説明図。

【図 10】

1 つのクラスと、それに対する設計者の意図する状態の定義の有無とそれに応

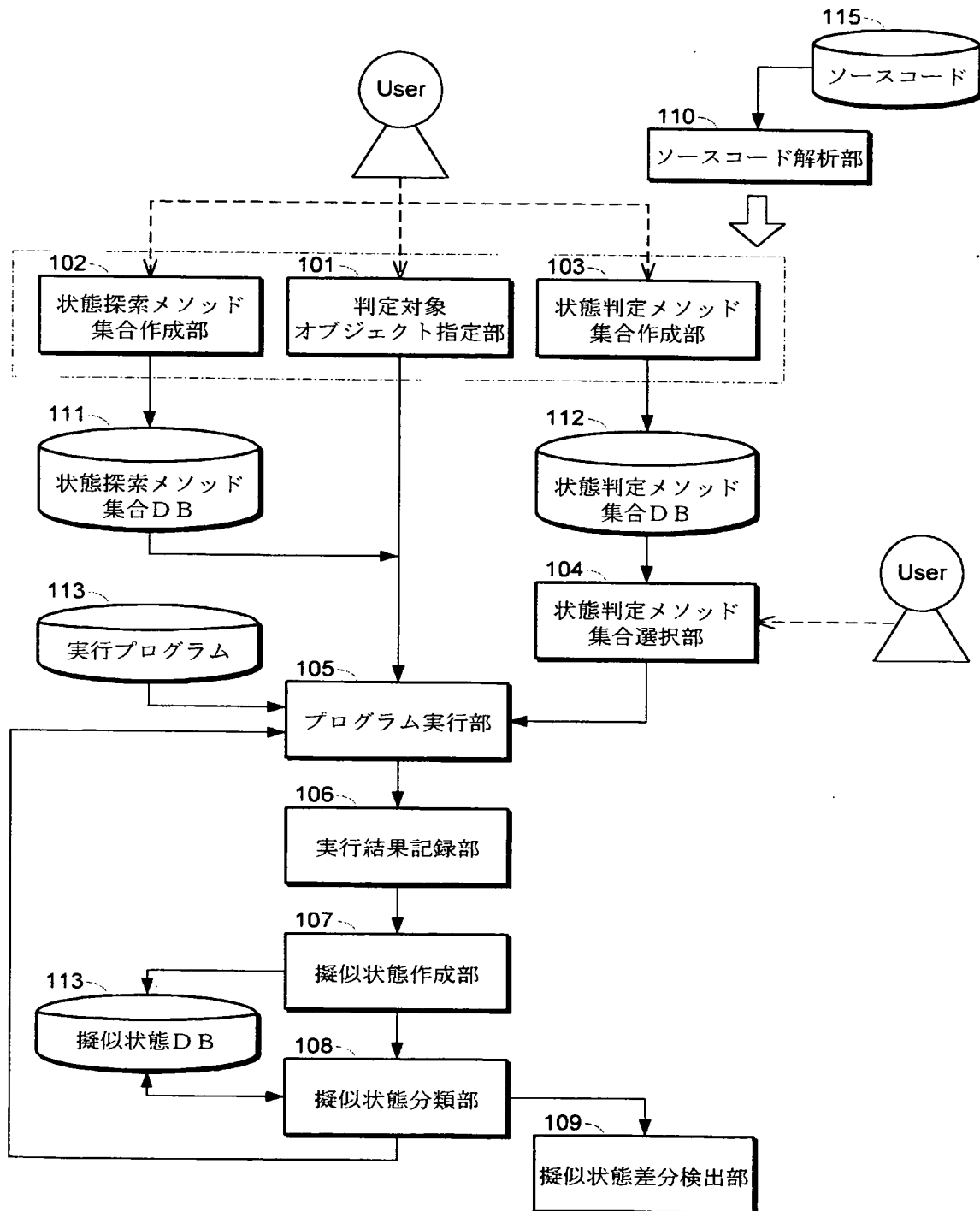
じたテストケースのリストの一例を示す説明図。

【符号の説明】

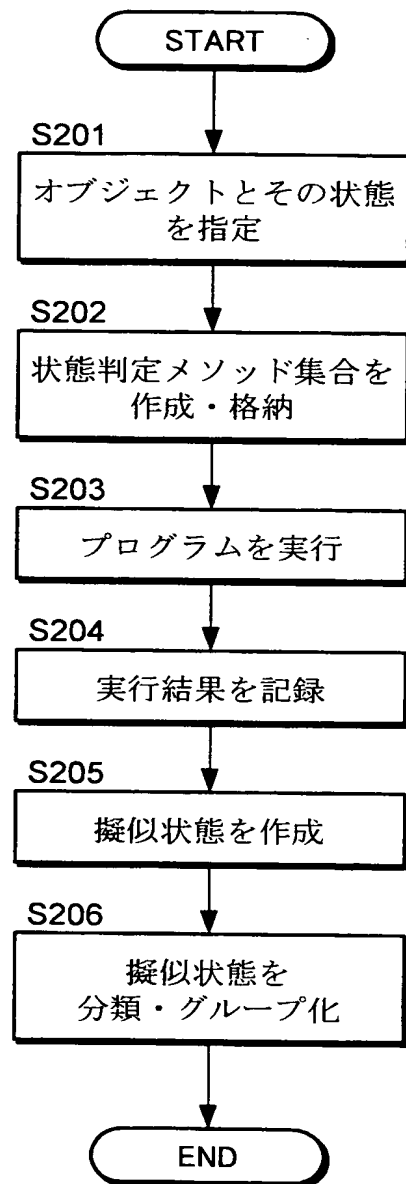
- 1 0 1 …判定対象オブジェクト指定部
- 1 0 2 …状態探索メソッド集合作成部
- 1 0 3 …状態判定メソッド集合作成部
- 1 0 4 …状態判定メソッド集合選択部
- 1 0 5 …プログラム実行部
- 1 0 6 …実行結果記録部
- 1 0 7 …擬似状態作成部
- 1 0 8 …擬似状態分類部
- 1 0 9 …擬似状態差分検出部
- 1 1 0 …ソースコード解析部
- 1 1 1 …状態探索メソッド集合データベース
- 1 1 2 …状態判定メソッド集合データベース
- 1 1 3 …実行プログラム
- 1 1 4 …擬似状態データベース
- 1 1 5 …ソースコードデータベース

【書類名】 図面

【図 1】



【図 2】



【図 3】

```
//
class Door
{
protected:
    int    angle;
    bool   flgLock;

public:
    void    setAngle(int a);
    void    lock();
    void    unLock();
    bool    isOpened();
};

void Door::setAngle(int a)
{
    if (flgLock){
        return;                // ブロック 1 1
    }

    if (a < 0){
        angle = 0;            // ブロック 1 2
    } else if (a <= 90){
        angle = a;            // ブロック 1 3
    } else {
        angle = 90;            // ブロック 1 4
    }
}

void Door::lock()
{
    if (angle){
        angle = 0;            // ブロック 2 1
    }
    flgLock = true;           // ブロック 2 2
}

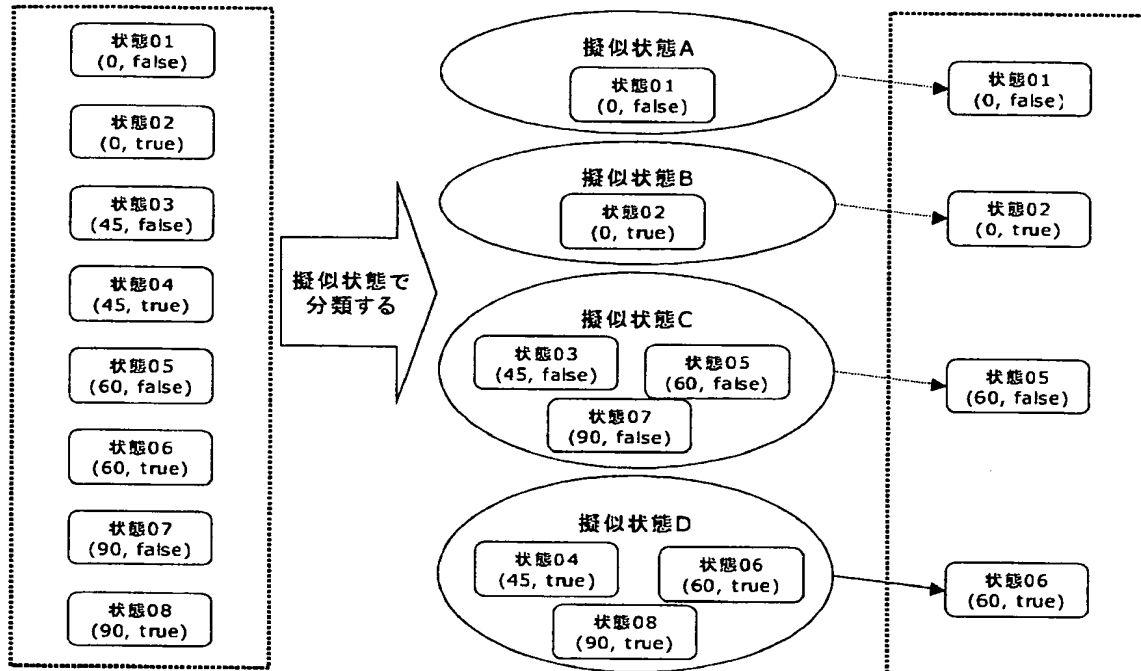
void Door::unLock()
{
    flgLock = false;           // ブロック 3 1
}

bool Door::isOpened()
{
    if (angle == 0){
        return false;         // ブロック 4 1
    } else {
        return true;          // ブロック 4 2
    }
}
//
```

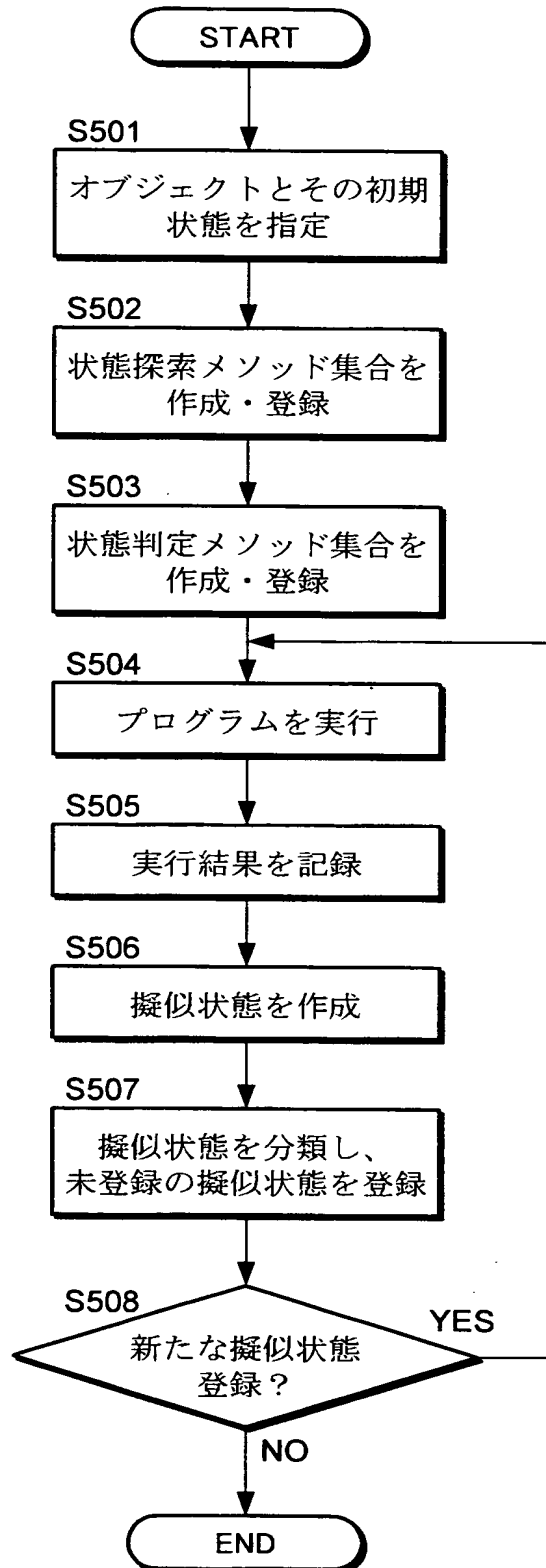
【図4】

ユーザが指定した状態群

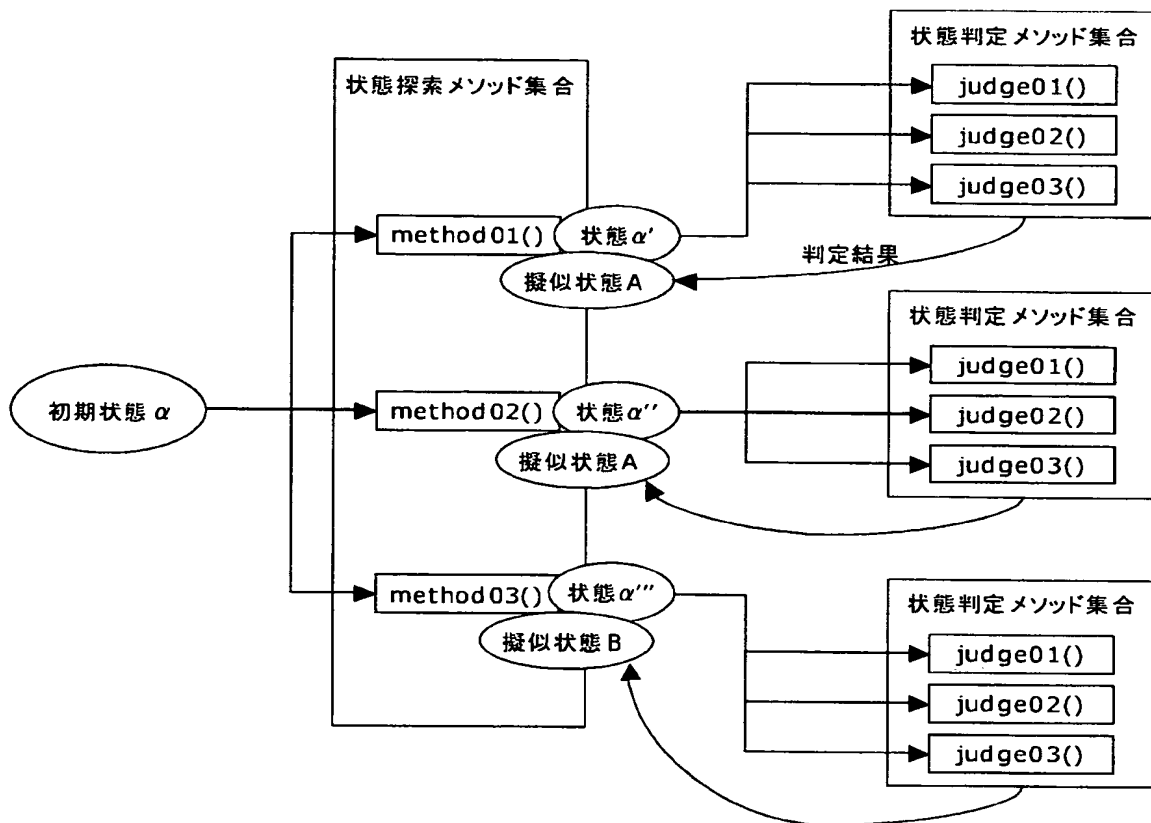
各擬似状態の代表元を抽出



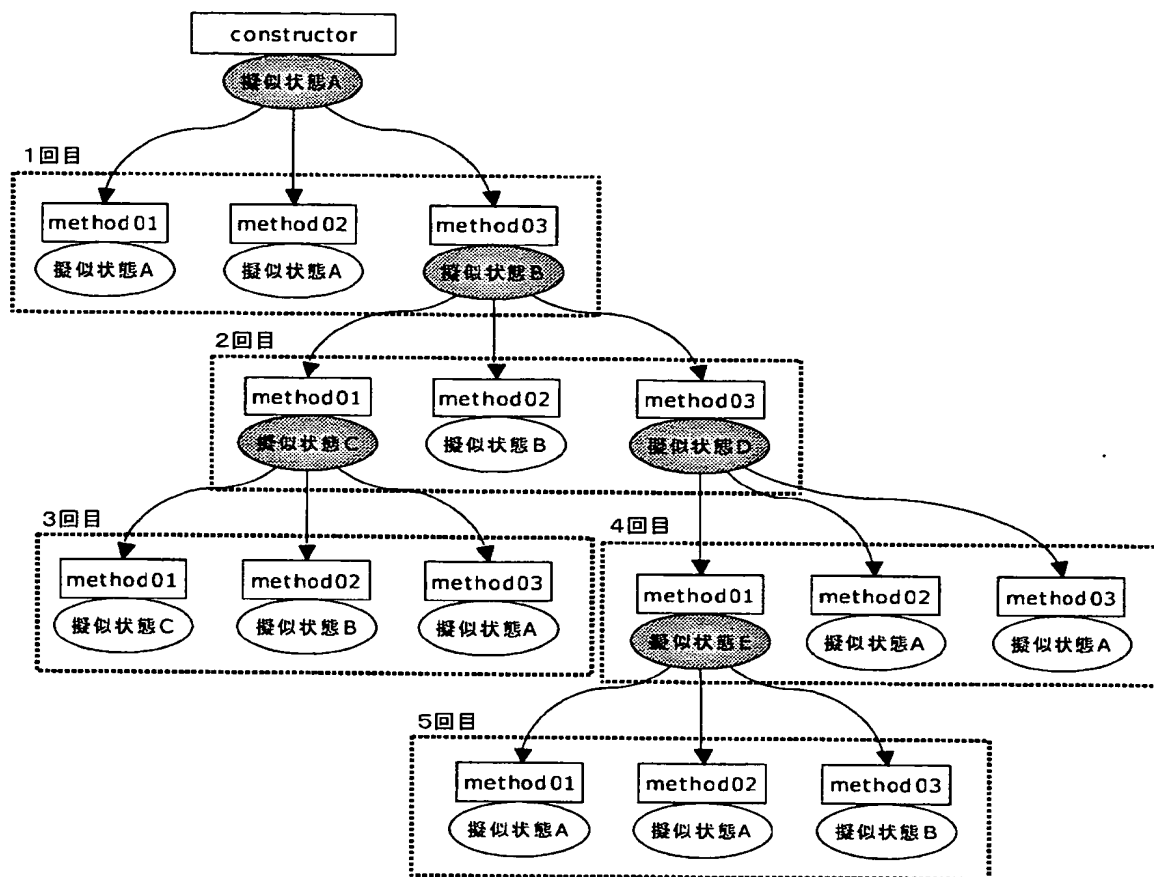
【図 5】



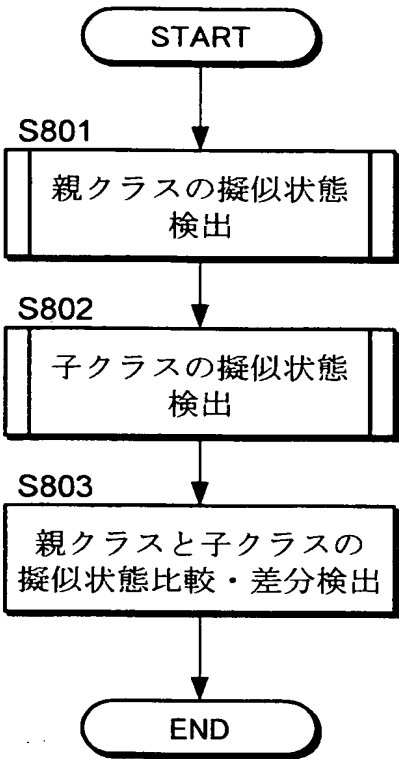
【図 6】



【図7】



【図 8】



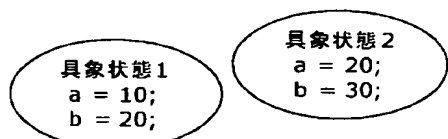
【図 9】

<pre>class Parent { int p1; int p2; void PMethod01(); void PMethod02(); void PMethod03(); }; class Child : public Parent { void CMethod01(); void CMethod02(); };</pre>				
インスタンス	初期状態	状態探索メソッド集合	状態判定メソッド集合	発見された 擬似状態
Parent	p1 = 0, p2 = 0	{ PMethod01(), PMethod02(), PMethod03() }	{ PMethod01(), PMethod02(), PMethod03() }	擬似状態A 擬似状態B
Child	p1 = 0, p2 = 0	{ CMethod01(), CMethod02(), }		擬似状態A 擬似状態B 擬似状態C

【図 10】

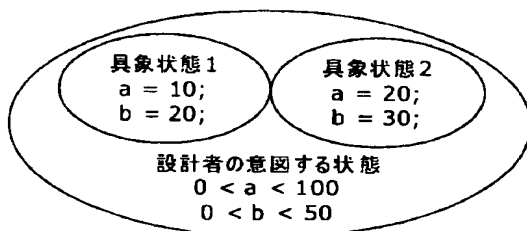
```
class DUMMY
{
    int a;
    int b;
    void methodA();
    void methodB();
};
```

1. 設計者が意図する状態が定義されていない場合のテスト
具体的な状態を設定し、その状態から様々なメソッドを呼び出す



テストケース 1 : 具象状態 1 から methodA を呼び出す
テストケース 2 : 具象状態 1 から methodB を呼び出す
テストケース 3 : 具象状態 2 から methodA を呼び出す
テストケース 4 : 具象状態 2 から methodB を呼び出す

2. 設計者が意図する状態が定義されている場合のテスト
設計者の意図する状態に対して、具体的な状態を作り出し、その状態から様々なメソッドを呼び出す



テスト効率化のために差し当たっては
「設計者の意図する状態」の1つの具体例である具象状態 1 をもとにテストすればよい。

テストケース 1 : 具象状態 1 から methodA を呼び出す
テストケース 2 : 具象状態 1 から methodB を呼び出す

【書類名】 要約書

【要約】

【課題】 クラスの状態の明確な定義の有無に拘わらず、あらゆるクラスに対して、「状態の網羅」という 1 つの尺度に基づいたテストの効率化を可能にする。

【解決手段】 判定対象オブジェクトとその判定対象状態を指定し（S 2 0 1）、状態判定メソッド集合を作成、登録する（S 2 0 2）。複数の状態判定メソッド集合が作成された場合には、使用する状態判定メソッド集合を選択する。判定対象状態である各具象状態に対して、選択された状態判定メソッド集合に含まれる全てのメソッドをそれぞれ呼び出し（S 2 0 3）、実行結果を記録する（S 2 0 4）。各具象状態に対して、プログラムの実行に使用した状態判定メソッド集合と実行結果とを足して擬似状態を作成する（S 2 0 5）。複数の擬似状態を得た複数の異なる具象状態を、同一の擬似状態となる具象状態ごとにグループ化する（S 2 0 6）。

【選択図】 図 2

特願 2002-378218

出願人履歴情報

識別番号

[000003078]

1. 変更年月日

2001年 7月 2日

[変更理由]

住所変更

住 所

東京都港区芝浦一丁目1番1号

氏 名

株式会社東芝